

DTIC FILE COPY

①

219800-13-T(1)

Technical Report

MaxVideo NEIGHBORHOOD
PROCESSOR PIPELINE BOARD
TEST SOFTWARE DESCRIPTIONS

AD-A227 618

J. JONIK

JULY 1990

Prepared for:

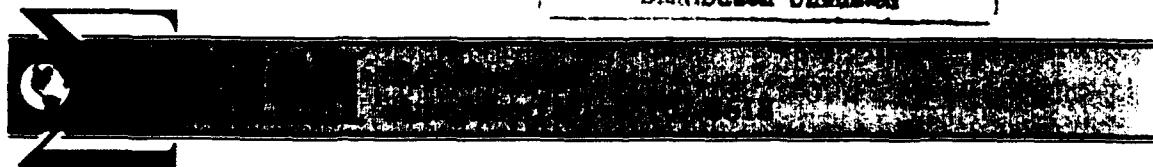
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-5800

Contract No. 42-3638

DTIC
ELECTE
OCT 04 1990
S B D

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited



90 10 04 008

1. Report No. 219800-13-T(1)		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle MaxVideo Neighborhood Processor Pipeline Board Test Software Descriptions				5. Report Date July 1990	
				6. Performing Organization Code ERIM	
7. Author(s) J. Jonik				8. Performing Organization Report No. 219800-13-T(1)	
9. Performing Organization Name and Address Environmental Research Institute of Michigan P.O. Box 8618 Ann Arbor, MI 48107-8618				10. Work Unit No. PTD-90-063	
				11. Contract or Grant No. 42-3638	
				13. Type of Report and Period Covered Software Descriptions Sept. 1989-Oct. 1990	
12. Sponsoring Agency Name and Address Sandia National Laboratories P.O. Box 5800 Albuquerque, NM 87185-5800				14. Sponsoring Agency Code	
15. Supplementary Notes					
16. Abstract <p>This document contains descriptions of the tests that implement the MVNPP High-Level Test Software Specification. The specification was written in order to provide a description of tests that exercise the functionality of the MVNPP board. The MVNPP testing is divided into two main sections; VMEbus testing verifies the communication of the MVNPP over the VMEbus, and the Datacube interface testing checks image and stagecode transfers over the ROI bus, and stagecode transfers over the coefficient loading bus. Several of the Datacube boards are used when executing the tests.</p> <p>The description for each test provides the calling procedure, any input or output files necessary (stagecode, images, or results files), and errors that may be flagged. It is beyond the scope of this document to give a full description of each phase of a test and its purpose.</p> <p>The go/nogo test that checks the functionality of a given system including an MVNPP and other Datacube boards is also described in this document. The MVNPP stages are programmed over the coefficient loading bus, and then an image is transferred from the Datacube boards to the MVNPP and back. The resultant image is then checked. (KR) ←</p>					
17. Key Words Software			18. Distribution Statement Approved for public release; distribution is unlimited.		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page)		21. No. of Pages 10	
				22. Price	

CONTENTS

1.0	INTRODUCTION	1
2.0	LOADING SOFTWARE FROM TAPE	2
3.0	CREATING C4PL STAGECODE AND IMAGE FILES	3
4.0	VMEBUS TESTS	4
4.1	Register Write/Read Test	4
4.2	Pipeline Sizing Test	5
4.3	FIFO Errors Test	5
5.0	DATAcube INTERFACE TESTS	6
5.1	Hardware Configuration for Datacube Testing	6
5.2	Image Circulation Test	7
5.3	Coefficient Loading Bus Test	7
6.0	OTHER MAIN PROGRAMS PROVIDED	8
6.1	Go/NoGo Test	8
6.2	Demonstration Program	9
6.3	Make .LNOC File	9

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

1.0 INTRODUCTION

This document contains descriptions of the tests that implement the MaxVideo (tm) Neighborhood Processor Pipeline (MVNPP) High-Level Test Software Specification (IPTL-90-089). The tests fall under two categories: VMEbus-based testing and Datacube (tm) interface testing. All testing is done from a Sun-3 using a CYTO-HSS workstation or an attached VME card cage.

Three other main programs are also described in this document. They include the go/nogo test, a demonstration program, and a tool to convert stagecode files to the correct format.

This specification assumes the reader has read and is familiar with the following manuals:

MVNPP Programmer's Manual (IPTL-90-014)
Stage Programmer's Manual (IPTL-89-294)
Datacube's ROI-STORE Manual

Datacube, MaxVideo, ROI-STORE, DIGIMAX, and MAX-MUX are registered trademarks of Datacube Incorporated.

2.0 LOADING SOFTWARE FROM TAPE

The following directories and files are provided on the distribution tape:

```
mains/mvnpptest/imagecirc  
mains/mvnpptest/clbusprog  
mains/mvnpptest/gonogo  
mains/mvnpptest/makelnoc  
mains/mvnpptest/demo
```

```
mains/vmetest/regrw  
mains/vmetest/pipesize  
mains/vmetest/fifoerrors  
mains/vmetest/utils
```

```
mvnpp/reglevel  
mvnpp/lowlevel  
mvnpp/toplevel
```

```
roimath/getROIparams.c  
roimath/roitime2.c  
roimath/Makemathlib
```

```
include/mvnppbits.h  
include/mvnppstageop.h
```

Use the following commands to read the files:

```
cd maxtools  
  
tar xvf /dev/rst8
```

3.0 CREATING C4PL STAGECODE AND IMAGE FILES

Image files used by the test software are created in C4PL (tm) using the 'SAVE' command. The first 512 bytes of the image file are header information and are not used when transferring images through the pipeline.

Stagecode files created in C4PL are created using the C4PL 'STORECODE' command. These files are referred to by the MVNPP software as .NOC (Neighborhood Object Code) files. The 'makeLnoc' program found under the 'mains/mvnppptest/makelnoc' directory takes this type of file as input and produces a .LNOC file which is expected by the test programs. This conversion is necessary because the 'STORECODE' command does not make use of the 31 registers available in the chip stage. The 'mvnppLoadCmds' routine issues a long program command to the stages which involves the 31 chip format registers. The conversion from .NOC to .LNOC stagecode files may also be done from within a program by calling 'mvnpp/toplevel/mvnppCnvtOp', and providing file pointers to the input and output files. The provided routine, 'mvnppReadCode', reads in stagecode from a file of the .LNOC format.

C4PL is a registered trademark of the Environmental Research Institute of Michigan

4.0 VMEBUS TESTS

The VMEbus tests are run from within C4PLSIM. The MVNPP board may be tested from a CYTO-HSS workstation or a VME card cage attached to a Sun workstation. The base address of the MVNPP must be entered; this will be 0x801300 in most cases (the default address). See the MVNPP Programmer's Manual (IPTL-90-014) section on MVNPP Configuration for instructions on finding the board's register base address. Once in C4PL, enter 'b 0x801300' (or another address if the base is not this one).

If working on a Sun workstation, run C4PLSIM and select '1' from the prompt to indicate that the MVNPP hardware is to be used. The virtual base address of the board is necessary. Type 'print cont_config' for a list of addresses of hardware that is present, then use the 'b' command to set the address of the MVNPP. Once the base is set, the tests described in the next sections may be run.

4.1 Register Write/Read Test

The register write/read test has 4 phases that are described in the Test Specification document (IPTL-90-089). The following command, entered at the C4PL prompt, runs the test:

```
C4PL > REGRW n
```

where 'n' specifies the phase to execute. Eliminating the 'n' causes all phases of the test to be executed. If a value read back from a register is not the expected one, a message is given indicating the expected and received values.

4.2 Pipeline Sizing Test

The pipeline sizing test returns the number of stages found in the pipeline. This test has only one phase. Since the number of stages available on an MVNPP board varies, the test does not expect a specific result. The number of stages found is displayed and it is up to the user to verify whether it is correct. If the pipeline is broken, a message is given instead of the number of stages found. The following command, entered at the C4PL prompt, runs the test:

```
C4PL > PIPESIZE
```

4.3 FIFO Errors Test

The FIFO errors test has 1 phase as described in the Test Specification document. The following command, entered at the C4PL prompt, runs the test:

```
C4PL > FIFOERROR
```

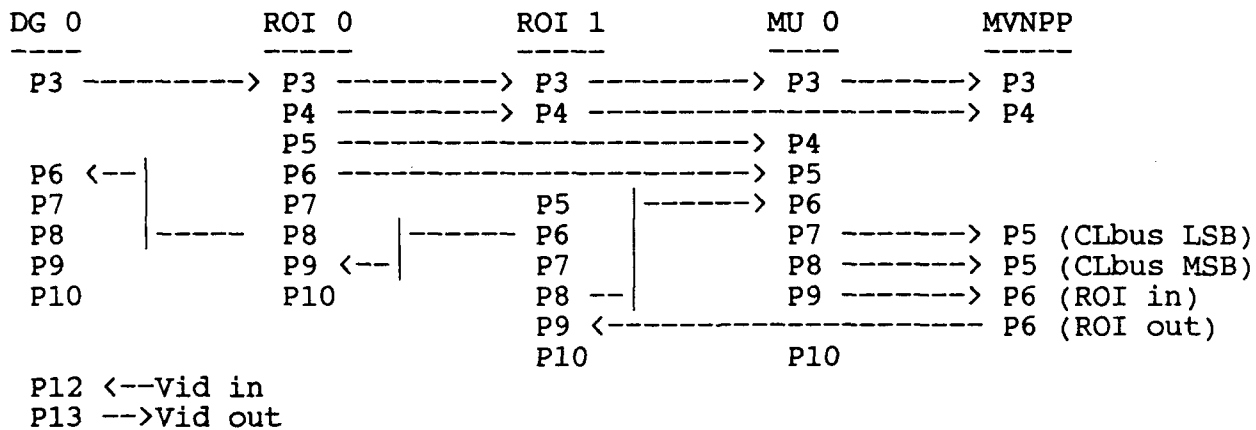
If a FIFO error is reported from the MVNPP when there shouldn't be one, or if a forced over or underrun does not produce an error, an appropriate error message is given.

5.0 DATACUBE INTERFACE TESTS

The 'mains/mvnpptest' directory contains two subdirectories, 'clbusprog' and 'imagecirc', which contain programs for testing the Datacube interface. The configuration for the hardware is explained in the next section, followed by the test procedures for each.

5.1 Hardware Configuration for Datacube Testing

The following configuration should be used for all of the Datacube interface testing. All tests do not use all boards; however, this single configuration was developed to simplify setup.



DG - DIGIMAX board

MU - MAX-MUX board

5.2 Image Circulation Test

The image circulation test has 14 phases as described in the MVNPP Test Software Specification. All stages are deactivated for the test, so no stagecode is necessary. The images used are randomly created at run time. To execute the test, enter the following:

```
imagecirc [p] [p] ... [p]
```

where '[p]' is a phase to run. Eliminating the phase numbers causes all phases of the test to be executed. Messages are given as each phase is run, and any errors that occur are reported. Since all stages are deactivated, a test is passed if the image that has been circulated from the ROI-STORE to the MAX-MUX to the MVNPP and back to the ROI-STORE matches the original image, and no errors are reported by the MVNPP. If the images do not match, a message is given for each discrepancy found along with the expected and actual values.

5.3 Coefficient Loading Bus Test

The coefficient loading bus test has 7 phases as described in the MVNPP Test Software Specification. The stagecode used in the test is randomly created at run time. The 256x256 image required for the seventh stage is a ramp image that is also created at run time. Entering the following command executes the test:

```
clbusprog [p] [p] ... [p]
```

where '[p]' is the phase to run. Eliminating the phase numbers causes all phases of the test to be executed. For each phase, a readback of the registers and memory is done, and any discrepancies found are reported. The seventh phase also circulates an image through stages programmed with nop stagecode. If the image is altered in any way, an error message is given.

6.0 OTHER MAIN PROGRAMS PROVIDED

6.1 Go/NoGo Test

This test verifies the functionality of the MVNPP board(s). It is a pass/fail type of test. Stagecode and images are read from files; the provided files may be used, or new ones may be created in C4PL as described in section three. A loop is entered that will program the stages and transfer the image through the pipeline; this is done until all stagecode has been programmed into the stages. The resultant image is then verified with an expected result image that can be created from running the same stagecode and initial image in C4PL (or the provided file may be used). If any discrepancy between the images is found, it will be reported and the test marked failed. The program expects the image file size to be 512 by 512.

At the beginning of the test, the user is prompted for the input image file, the expected result image file to use for the comparison, and the stagecode file. The program produces a file called "results" which contains the image resulting from the test. This allows for a comparison of the image files after the test has been executed or examination of failure results. The following files have been provided for running the test:

- airport.img - 512 by 512 input image
- foa.lnoc - stagecode file
- airport.cmp - resultant image comparison file

The test can be run by entering 'gonogo' at the shell prompt. The program then prompts for the three file names. If the image resulting from the test does not match the comparison image, a message is given for each pixel discrepancy along with the expected and actual values.

6.2 Demonstration Program

A program had been created for demonstration purposes, and it has been included to provide an example. The program first reads in a stagecode file (DEMO.LNOC) and an image file (DEMO.IMG). Four stages are then programmed and a loop is entered that circulates the image through the pipeline 200 times. The stages' registers and memory are read back and verified after the programming. Any discrepancy found is reported. There is no verification of the resultant image. Enter 'demo' to run the program.

6.3 Make .LNOC File

The program 'makeLnoc' is a conversion utility that produces .LNOC format files from .NOC files (created in C4PL with the 'SAVE' command). The program expects the names of an input and output file on the command line. 'temp.noc' and 'temp.lnoc' are used as defaults if a name is not given. The files are opened and created, respectively. Then, 'mvnppCnvtOp' is called. The stagecode in the .NOC file is read in, converted to the .LNOC format and written to the .LNOC file. This conversion is necessary because the C4PL command to save stagecode does not make use of the 31 registers available on the chip stages. The provided MVNPP routines work with long program load commands which require the extended register set. The program also checks whether the PRAM or NRAMs should be bypassed, and if so, the appropriate bits are set in the stage bypass register. Refer to the Stage Programmer's Manual for more information about the registers. The routine provided to read stagecode in from a file (mvnppReadCode) requires files in the .LNOC format. Enter the following to run this conversion utility:

```
makeLnoc in_file.noc out_file.lnoc
```

where 'infile.noc' is the name of the stagecode file saved from C4PL, and 'outfile.lnoc' is the name of the output file where the converted stagecode is to be stored.